übersicht binärcode:

bit	kleinste informationseinheit	1 zustand
byte	8 bit	256 (2^8... kombinatorik) zustände (in der informatik mit 0..255 durchgezählt)
kByte	1024 byte	2^{10}=1024, abweichung zum alltag: k=1000
MByte	1024 kByte = 1024^2 byte	wie bei k erneut abweichung zum alltag. daher immer die abweichungen zw. MB und kB ;)

a1) rechne die binärzahlen in dezimalzahlen um! (für schnelle: auch in hexzahl umrechnen)

1101 1111 223 (=128+64+16+8+4+2+1, da diese 2er-potenzen auf 1 stehen).
einfacher: 255 - 32 = 223. es fehlt ja "nur" die 2^5, also die 32 ;)

0011 0011 51

0101 0101 85

für schnelle: hexzahlen sind zahlen mit der basis 16. es gibt 1er, 16er, 16^2er (256er) usw.
ein byte kann man bereits mit zwei hexzahlen darstellen, da man bis zu 15 1er
und bis zu 15 16er zur verfügung hat. um "zweistellige" zahlen zu haben,
gibt es eine neue buchstaben-notation: 10=A, 11=B, 12=C, 13=D, 14=E, 15=F.
damit sind zahlen zwischen 0 (00) und 255 (FF=15*16+15*1) darstellbar!

das umrechnen von binär -> hex ist für ein byte sehr einfach, denn 2^4=16 :)
man wertet einfach die ersten 4 bit aus und dann die hinteren 4.

1101 1111 der vordere 4bit-block ist dezimal 13 (=8+4+1), der hintere ist 15.
damit ist die hexzahl "13" "15", was man aber mit buchstaben als DF schreibt!

0011 0011 "3" "3" -> hexzahl 33.

für ganz schnelle: um von dez auf hex zu kommen, kann man den euklidischen algorithmus
(s.u.) anwenden: die zahl 1101 1111 war dezimal 223. also...

223	:16	=	13	Rest	15 (in hex: F)
13	:16	=	0	Rest	13 (in hex: D)

liest man von unten nach oben (DF), hat man das ergebnis von oben ;)
D=1101 und F=1111 und schon hat man wieder die binärdarstellung. passt alles.

a2) rechne die drei dreistelligen dezimalzahlen um in binärzahlen! (für schnelle: auch in hexzahl)

101 (einhundertundeins) 0110 0101 entweder mit dem euklidischen algorithmus (s.u.) oder
durch probieren.
für schnelle: das ist 65 als hexzahl (hex=abkürzung für hexadezimal),
denn man wertet wieder die 4 bits vorne (0110) bzw. hinten (0101) aus.

111 (einhundertelf) 0110 1111

110 (einhundertzehn) 0110 1110

a3) nutze für a2) den euklidischen algorithmus!

bespiel: erste dezimal zahl einhundertundeins (101)

101	:2	=	50	Rest	1
50	:2	=	25	Rest	0
25	:2	=	12	Rest	1
12	:2	=	6	Rest	0
6	:2	=	3	Rest	0
3	:2	=	1	Rest	1
1	:2	=	0	Rest	1

man kommt irgendwann bei 0 Rest 0 oder 0 Rest 1 an. hier stoppt der algorithmus und
man liest die binärzahl von UNTEN NACH OBEN, also hier 1100101.

in byteschreibweise muss man vorne noch eine 0 anfügen, damit wir unsere 8 stellen haben.
woher soll euklid auch unsere 8-stellige standardschreibweise kennen ;)

a4) addieren von binärzahlen ist einfach! bsp: 110 (in dezimal 4+2=6) und 100 (dezimal: 4).

```
        110 + 100 = 1010        sollte stimmen, denn 6+4=10
                                und 10 ist binär: 1010 (ein 8er,kein 4er,ein 2er,kein 1er)

        wie man das rechnet? schriftliches addieren aus der grundschule hilft:

                 110
        +        100
        ------------
                1010    schrittweise von hinten nach vorne: 0+0=0. 1+0=1. dann kommt 1+1,
                        was im binärsystem bereits einen übertrag erzwingt!
                        [bspw. so, also ob du 5+7 im dezimalsystem vorliegen hast;
                        hier schreibt man ja dann auch "2 übertrag 1"]

                        es gilt hier: 1+1=0 übertrag 1! so entsteht die "10" zu beginn.
```

a5) multiplikation von binärzahlen ist auch einfach! bsp: 111 (dez. 7) und 101 (dez. 5).

```
        111 * 100 = 100011 sollte stimmen, denn 7*5=35 und das ist ein 32er, ein 2er und ein 1er.

        wie man das rechnet? schriftliches multiplizieren aus der grundschule hilft:

        111 * 101       wir werten von hinten nach vorne bei der 101 aus:
        ---------
              111       1 hinten bei 101:       1*111 ist natürlich 111.
             000        0 mitte bei 101:        0*111 ist sehr einfach... 000.
            111         1 vorne bei 101:        1*111 ist immer noch 111.

        ---------       nun werden 3 binärzahlen addiert, wobei auf die einrückung zu achten ist.
           100011       es ergibt sich die richtige lösung (achtung: 1+1=0 übertrag 1 bedenken!)

        in unserer byteschreibweise von binärzahlen sind noch 00 vorne hinzuzufügen:

        0010 0011 ist dann die dezimalzahl 35 in byteschreibweise!
```